Read Me file - Classification and Topic Modeling tasks on textual datasets - Alberto Arienti and Sofia Turroni

Project description

The aim of this project is to perform text mining techniques on a set of data, following the machine learning pipeline from the pre-processing of the raw data to the training and evaluation of the models. The main objective is to classify the content into predefined categories, and extract the main themes (topic modeling) to obtain a structured representation of the information. Pre-processing includes cleaning, tokenization, and removal of irrelevant elements. Classification is implemented using machine learning algorithms, while the identification of main topics exploits topic modeling methods such as Latent Dirichlet Allocation (LDA). This approach makes it possible to transform unstructured data into structured, meaningful information.
Here are the main files:

- Colab notebook: *Project_Code.ipynb*
- Detailed PDF report: *Project_Report.pdf*
- PowerPoint Presentation: *Project_Presentation.pptx*

Data source

The dataset used is a part of the **PubMed 200k RCT** dataset, accessible by *Kaggle*, and available at the following link: PubMed. To develop the text analysis not the entire dataset have been used, but the part called `Train.csv` inside the folder `PubMed_20k_RCT`, which is a smaller versione of the original dataset, consisting in 20.000 scientific papers.
In the classification task the corpus used as explanatory variable is contained in the column *abstract_text*, while the target variable is the column *target*.
In the topic modeling task, the data have been grouped by value in column *abstract_id* and the text has been aggregated by sum function.

Python libraries

The code developed for all the steps of the project can be found in file *Project_Code.ipynb*, there are not specific requirements on the environment's set up. We are following presenting the necessary libraries to correctly run the code. For text Pre-Processing:

```
import nltk
import re
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import WordPunctTokenizer
```

```
import string
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger_eng')
stop_words = nltk.corpus.stopwords.words('english')
wnl = WordNetLemmatizer()
```

For Text Classification:

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, recall_score,
    classification_report, precision_score
```

For the Topic Modeling:

```
pip install pyLDAvis
import matplotlib.pyplot as plt
import gensim
import numpy as np
import spacy
from gensim.models import CoherenceModel, LdaModel, LsiModel, HdpModel
from gensim.corpora import Dictionary
from sklearn.metrics.pairwise import cosine_similarity
import pyLDAvis.gensim
import pyLDAvis.gensim_models as gensimvis
import os, re, operator, warnings
warnings.filterwarnings('ignore')
%matplotlib inline
import random
```

Results

The main findings of the analysis are summarized in the PDF report under *Project_Report.pdf*. Key insights and visualizations are presented in the PowerPoint presentation *Project_Presentation*.